A Fuzzy Logic Controller for Speed Control of a DC Series Motor Using an Adaptive Evolutionary Computation

Gi Hyun Hwang, Hyun-Joon Hwang, Dong-Wan Kim, and June Ho Park

Abstract : In this paper, an Adaptive Evolutionary Computation(AEC) is proposed. AEC uses a genetic algorithm(GA) and an evolution strategy (ES) in an adaptive manner in order to take merits of two different evolutionary computations: global search capability of GA and local search capability of ES. In the reproduction procedure, proportions of the population by GA and ES are adaptively modulated according to the fitness. AEC is used to design the membership functions and the scaling factors of fuzzy logic controller (FLC). To evaluate the performances of the proposed FLC, we make an experiment on FLC for the speed control of an actual DC series motor system with nonlinear characteristics. Experimental results show that the proposed controller has better performance than that of PD controller.

Keywords : adaptive evolutionary computation, fuzzy logic controller, DC series motor

I. Introduction

During the last decade, fuzzy logic control has been attracted great attention from both the academic and industrial communities. Recently, fuzzy logic controller has been suggested as an alternative approach to conventional control techniques for complex control system, such as nonlinear or time delay system. That is, the design of fuzzy logic controller (FLC) does not require a mathematical description of the control system and the fuzzy controller can compensate the environmental variation during the operating process [1-3].

However, we cannot obtain good control performances if the membership functions, fuzzy rules and scaling factors are incorrect. Recently, the membership functions, fuzzy rules and scaling factors are determined by evolutionary computations (ECs), which are the probabilistic search methods based on genetics and evolutionary theory [4-5].

ECs are optimization algorithms based on the principles of the genetics and natural selection. There are three broadly similar avenues of investigation in the ECs: genetic algorithm (GA), evolution strategy(ES), and evolutionary programming (EP) [4-6]. When applied for solving the practical problems, each begins with a population of contending trial solutions brought to the task at hand. New solutions are created by randomly altering the existing solutions by the EC operation. The objective measure of performance is used to assess the fitness of each trial solution and selection.

It is obvious from the start that finding good settings for the EC parameters for a particular problem is not a trivial task. Several approaches are proposed. As One approach, adapting population size, crossover rate, and mutation rate are used. Arabas[7] proposed an adaptive method for maintaining variable population size. Schlierkamp-Voosen[8] presented a competition scheme, which dynamically allocates the number of trials given to different search strategies. The competition scheme changes not only the size of the subgroups, but also that of the whole population. Srinivas[9] proposed the Adaptive Genetic Algorithm(AGA), that is, the probabilities of crossover and mutation are varied depending on the fitness values of the solutions to maintain the diversity in the population and to sustain the convergence capacity of the GA.

The other approach is involved in 1) adapting the probabilities of crossover and mutation operator in the GA : the idea is that the probability of applying an operator is altered in proportion to the observed performance of the individual created by this operator. 2) Mutation parameters are adapted during the run in the ES. Hinterding[10] proposed Gaussian mutation operators for the GA, which allows the GA to vary the mutation strength during the run. Spears[11] proposed an adaptive mechanism for controlling the use of crossover in the ECs and explored the behavior of this mechanism in a number of different situations.

In this paper, the new methodology of evolutionary computations-an Adaptive Evolutionary Computation(AEC)-is proposed. AEC uses a GA and an ES in an adaptive manner in order to take merits of two different evolutionary computations: global search capability of GA and local search capability of ES. In the reproduction procedure, proportions of the population by GA and ES are adaptively modulated according to the fitness. AEC is used to design the membership functions and the scaling factors of the FLC. The proposed FLC is applied to the speed control of an actual DC series motor system with nonlinear characteristics.

II. Adaptive evolutionary computation

1. Motivation

In general, GA is known to offer significant advantages over traditional optimization methods. The most important ones are: the population-based search, the balance between exploitation(convergence) and exploration(diversity). But GA can suffer from the excessively slow convergence before providing an accurate solution because of its not exploiting local information. On the other hand, ES is well-known to exploit all local information in an efficient way. But, for problems with many local minima, it has the possibility of trapping in local minima.

In this paper, to reach the global optimum accurately and reliably in a short execution time, AEC bringing together the benefits of the GA and the ES is designed. In the AEC, GA operators and ES operators are simultaneously applied to the

Manuscript received : June. 14, 1999., Accepted : Jan. 3, 2000.
Gi Hyun Hwang, June Ho Park : School of Electronic, Electrical, and Communication Engineering, Pusan National University
Hyun-Joon Hwang : Dept. of Electricity, Pohang College
Dong-Wan Kim : Dept. of Electricity, Tong-Myong College

individuals of the present generation to create next one. Individuals with higher fitness value will have a higher probability of contributing one or more chromosomes in the next generation. This mechanism should give greater rewards to either the GA operation or the ES operation that produces superior offspring.

2. Adaptive evolutionary computation

In the AEC, the number of individuals created by the GA operation and the ES operation changes adaptively. Configuration of the AEC is shown in Fig. 1. In the AEC, the individual is represented as a real number chromosome, not a binary chromosome, which makes it possible to hybridize for the GA operation and the ES operation without loss of data. The main objective behind such implementation is that it enhances the performances of the AEC.

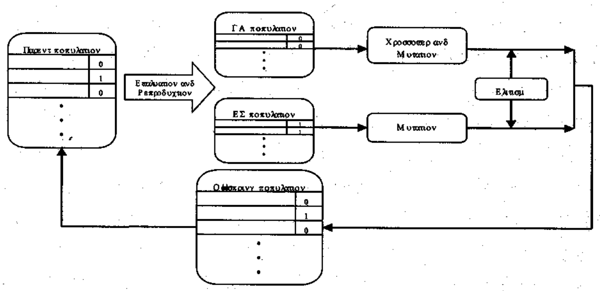


Fig. 1. Configuration of adaptive evolutionary computation.

ES forms a class of optimization techniques motivated by the reproduction of biological system and a population of individuals evolves toward the better solutions by means of the mutation and selection operation. In this paper, we adopted(μ, λ)-ES, that is, only the λ offspring by mutation operation competes for survival and the μ parents are completely replaced each generation. Also, self-adaptive mutation step sizes are used in the ES.

For the AEC to self-adapt its use of the GA and ES, each individual has an operator code in order to represent whether governed by the GA or the ES. Suppose a '0' refers to the GA, and a '1' to the ES. At each generation, if it is more proper to use the GA, more '0's should appear in the end of individuals. If it is more proper to use the ES, more '1's should appear. After reproduction by the roulette wheel selection according to the fitness, the GA operations - crossover and mutation - are performed on the individuals of which operator code is '0' and the ES operation, that is, mutation is performed on the individuals of which operator code is '1'. Elitism is also used. Best individual in the population is preserved to perform both the GA operations and the ES operation to next generation.

The major procedures of AEC are as follows:

1) *Initialization* : Initial population is randomly generated and operation code is randomly initialized for each individual. According to the operation code, GA operations are performed on the individuals with operator code '0', while ES operation is applied where the operator code is '1'.

2) *Evaluation and Reproduction* : Using the mate selection operator, individual chromosomes are selected in proportion to their fitness, which is evaluated using an objective function. After reproduction, GA operations(crossover and mutation) are performed on the individuals having an operator code of '0' and ES operation(mutation) is performed on those having an operator code '1'. At every generation, the percentages of '1's and '0's in the operator code indicate the performance of the GA and ES operator.

3) *Preservation of Minimum Number of Individuals* : At each generation, AEC may sometimes fall into a situation where the percentage of the offspring is nearly 100% or the offspring dies off. Therefore, it is necessary for the AEC to preserve a certain amount of the individuals for each EC operation. In this paper, we change the operation code of the individuals randomly with a high percentage, until the number of the individuals for each EC operation become higher than a certain amount of the individuals to be preserved. The predetermined minimum number of the individuals to be preserved is set to 20% of the population size.

4) *Genetic Algorithm* : The real-valued coding is used to represent a solution [4]. Simple crossover and uniform mutation are used as genetic operators.

5) *Evolution Strategy* : Only λ offspring by mutation operation competes for survival and μ parents are completely replaced each generation. Then, mutation is independently performed on each vector element by adding a normally distributed Gaussian random variable with mean zero and standard deviation (σ), as shown in (1). If the success ratio of mutation operator is smaller than the predetermined ratio, we increase the rate for standard deviation of mutation according to c_i, as shown in (2). If this ratio is larger than the predetermined ratio, we decrease the rate according to c_d, as shown in (2).

$$v_k^{t+1} = v_k^t + N(0,\sigma^t) \tag{1}$$

$$\sigma^{t+1} = \begin{bmatrix} c_d \times \sigma^t, & if \quad \phi(t) < \delta \\ c_i \times \sigma^t, & if \quad \phi(t) > \delta \\ \sigma^t, & if \quad \phi(t) = \delta \end{bmatrix} \tag{2}$$

Where, $N(0,\sigma^t)$: Vector of independent Gaussian random variable with mean of zero and standard deviations σ

V_k^t : k-th variable at t generation

$\phi(t)$: The success ratio of the mutation operator during the last t generation

c_d, c_i : The increase and decrease rates for the variance of the mutation

δ : Constants

III. Design of fuzzy logic controller using adaptive evolutionary computation

In designing the FLC, the exact mathematical modeling of control system is not needed and fuzzy rules can be represented as the knowledge of the experts. The design parameters used in this paper are given below.

- number of input/output variables : 2/1
- number of input/output membership functions : 7/7
- fuzzy inference method: max-min method

– defuzzification method: center of gravity

In this paper, the proposed method is to optimize the shapes of the membership functions and the scaling factors by the AEC. The general scheme is presented in Fig. 2. The input signals to the FLC are speed deviation (e) and the change in speed error(de). The output signals of the FLC are used to the speed control of an actual DC series motor system. Because the error and the change-of-error are used as input variables of the FLC, PD-like FLC is used. Rule base for the

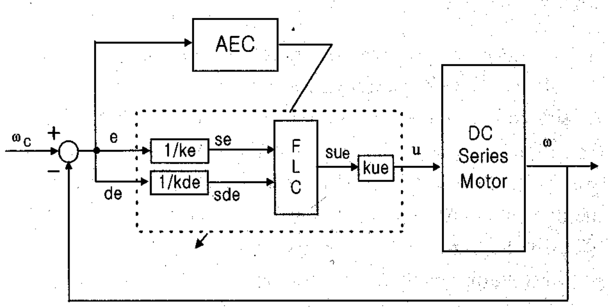


Fig. 2. Block diagram of fuzzy logic controller using the AEC.

PD-like FLC from the two-dimensional phase plane of the system in terms of the error and the change-of-error is shown in Table 1. The general approach to design the FLC is the division of the phase plane into two semi-planes, by means of switching-line. Within the semi-planes positive and negative control outputs are produced, respectively.

Table 1. PD-type fuzzy rules.

E \ De	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PM
ZE	NM	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PM	PM	PB
PM	NS	ZE	PS	PM	PM	PB	PB
PB	ZE	PS	PM	PM	PB	PB	PB

The magnitude of the output signals depends on the distance of the state vector from the switching line. When tuning the membership functions by the AEC, fuzzy rules are symmetric about switching line as shown in Table 1. In this table, linguistic variable NB means "Negative Big", NM "Negative Medium", NS "Negative Small", ZE "Zero", PS "Positive Small", PM "Positive Medium", and PB "Positive Big". The shape of the input and output membership function is assumed to be triangular. Also 7 input/output fuzzy sets for every input/output variable are used, hence the number of parameters of the FLC (center and width of the membership functions) is 63. But it takes long time for the AEC to tune 63 fuzzy parameters. In this paper, the ZE membership function is set at 0 and positive and negative membership function is symmetric about 0. So the number of parameters of the FLC is 21, that is, 3 centers and 4 widths for each variable, as shown in Fig. 3.

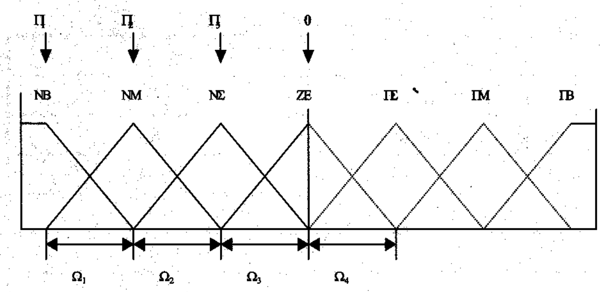


Fig. 3. Symmetrical membership functions.

Also the scaling factors of the FLC using the AEC are tuned, as shown in Fig. 2. To encode each parameter, real coding technique is used. String architecture for tuning the membership functions and the scaling factors is shown in Fig. 4. To evaluate each string in the population, the absolute error between output speed and reference speed of generator is used. The fitness function is defined in (3).

$$Fitness = \frac{1}{100 + \sum_{k=1}^{N} |\omega_c - \omega_k|} \tag{3}$$

Where, ω_k : actual speed

ω_c : desired speed

N : No. of data acquired during T second

String 1	P_{11}	···	P_{1l}	W_{11}	···	W_{1m}	SF_{11}	···	SF_{1k}
String 2	P_{21}	···	P_{2l}	W_{21}	···	W_{2m}	SF_{21}	···	SF_{2k}
⋮									
String n	P_{n1}		P_{nl}	W_{n1}		W_{nm}	SF_{n1}		SF_{nk1}

Where, n : population size

l : No. of center of the membership functions

M : No. of width of the membership functions

k : No. of scaling factors

Fig. 4. Strings architecture for tuning membership functions and scaling factors.

IV. Experimental results

Fig. 5 shows the speed control system structure of the speed control of an actual DC series motor. As shown in this fig., the AEC is used to optimize the shapes of the membership functions and the scaling factors. Fig. 6 represents the experiment apparatus for the speed control of the motor. Table 2 shows the simulation parameters of the AEC for tuning the FLC. Fig. 7 presents the shape of the membership functions by the AEC.

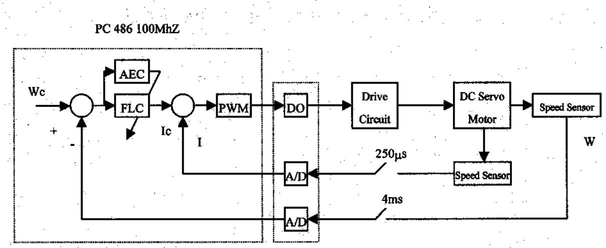


Fig. 5. Laboratory setup for dc series motor speed control.

Fig. 8 (a) provides the graphs of the fitness values by the

Table 2. Simulation parameters used AEC.

Methods	AEC
Size of population	20
Crossover probability	0.85
Mutation probability	0.05
δ	0.5
C_d	0.85
C_I	1.15
Number of Generation	20

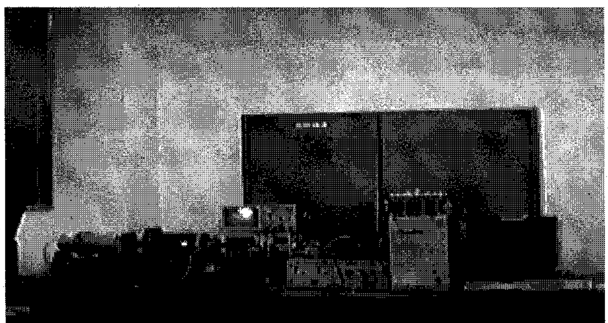
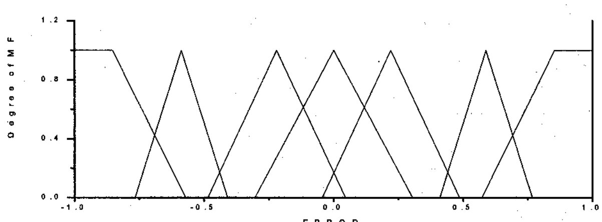
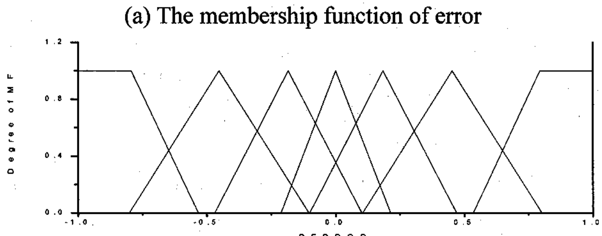


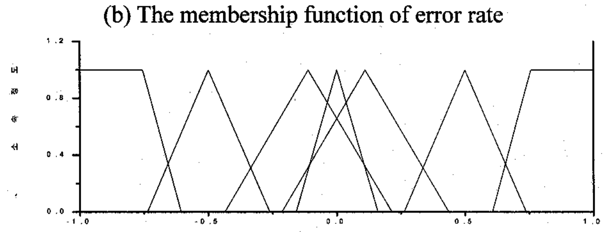
Fig. 6. Experiment apparatus of a DC series motor system.



(a) The membership function of error

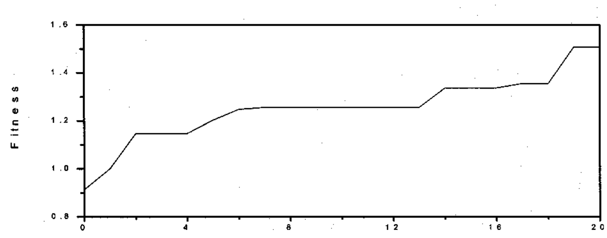


(b) The membership function of error rate

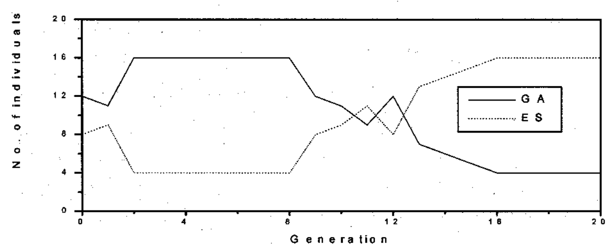


(c) The membership function of output

Fig. 7. Tuned membership functions using AEC.



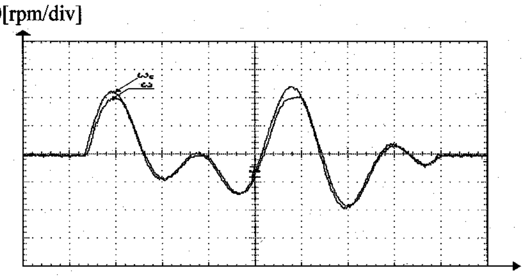
(a) Fitness value



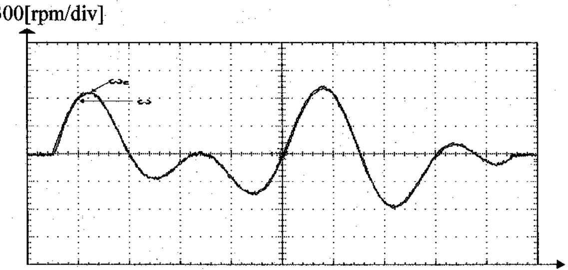
(b) Number of individuals of GA and ES in AEC

Fig. 8. Fitness functions and number of individuals of GA and ES.

AEC. Fig. 8 (b) provides the graphs of the number of individuals for the GA operations and the ES operation in the AEC. As shown in this fig., the percentage of individuals for the GA operation is greater than that of individuals for the ES operation in initial generation. But, from generation to generation, the percentage of individuals for the ES operation exceeds that of individuals for the GA operation. The AEC produces improved reliabilities by exploiting the "global" nature of the GA initially as well as the "local" improvement capabilities of the ES from generation to generation



(a) Speed response (PD Controller)



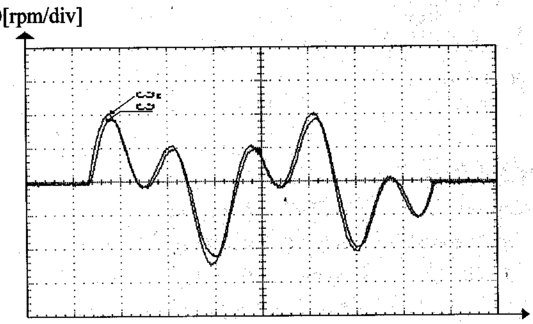
(b) Speed response (FLC)

Fig. 9. Comparisons of speed response with the PD controller and fuzzy logic controller.

Fig. 9 represents the experimental results of the DC series motor system for reference command used when tuning the FLC and the PD controller. As shown in this fig., speed response of the PD controller produces many differences between desired speed(ω_c) and actual speed(ω). Whereas, the proposed FLC produces more accurate speed response than the PD controller in terms of tracking performance.

Therefore, the proposed FLC demonstrates a better tracking performance as compared with the PD controller. To evaluate the robustness of the FLC, it is also tested over the reference command, which is not used when tuning. As shown in Fig. 10, experimental results confirm that the FLC shows the better performance over another reference command and various disturbances than that of the PD controller.

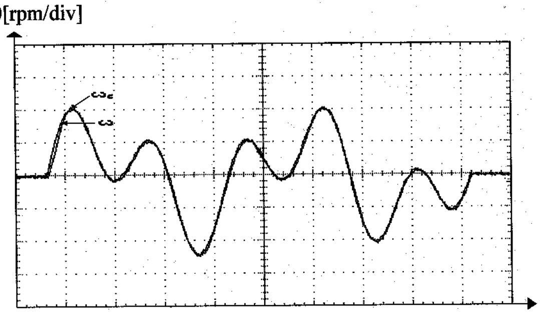
300[rpm/div]



2[sec/div]

(a) PD controller

300[rpm/div]



2[sec/div]

(b) Fuzzy logic controller

Fig. 10. Speed response with new reference speed.

V. Conclusions

In this paper, we have adaptively coupled the GA with the ES. The reason for combining the GA with the ES is that they compliment each other. ES will try to optimize locally, while the GA will try to optimize globally. In the AEC, GA operators and ES operators are simultaneously applied to individuals of the present generation to create next one. In the AEC, the number of individuals created by the GA operation and the ES operation changes adaptively. The AEC produces improved reliabilities by exploiting the "global" nature of the GA initially as well as the "local" improvement capabilities of the ES from generation to generation, so the AEC converges to the global optimal solution within a few generations.

The AEC is used to design the membership functions and the scaling factors of the FLC. The proposed FLC is applied to the speed control of an actual DC series motor system with nonlinear characteristics. Experimental results show that the FLC has the better control performance than the PD controller in terms of rising time and settling time. Also the FLC has the better performance over another reference command.

References

[1] Li-Xin Wang, "Stable adaptive fuzzy controllers with application to inverted pendulum tracking," *IEEE Trans. on Systems, Man, and Cybernetics-Part B : Cybernatics*, vol. 26, no. 5, pp. 677-691, Oct. 1996.

[2] Abraham Kandel and Gideon Langholz, "Fuzzy control systems", CRC Press, 1994Y. S. Kung and C. M. Liaw, "A fuzzy controller improving a linear model following controller for motor drives," *IEEE Trans. on Fuzzy Systems*, vol. 2, no. 3, pp. 194-201, Aug., 1994.

[3] D. E. Goldberg, *Genetic Algorithms in Search optimization, and Machine Learning*, Addison-Wesley publishing Company, INC., 1989.

[4] Z. Michalewicz, *Genetic Algorithms +Data Structures= Evolution Programs*, Springer-Verlag, 1994.

[5] Mitsuo Gen and R. Cheng, *Genetic Algorithms & Engineering Design*, A Wiley-Interscience Publication, 1997.

[6] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-a genetic algorithm with varying population size," *IEEE International Conf. on Evolutionary Computation*, pp. 73-78, 1994.

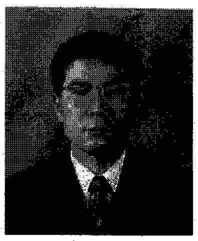
[7] D. Schlierkamp-Voosen and H. Muhlenbein, "Adaptation of population sizes by competing subpopulations," *IEEE International Conference on Evolutionary Computation*, pp. 384-389, 1995.

[8] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656-667, April, 1994.

[9] Robert Hinterding, "Gaussian mutation and self-adaptation for numeric genetic algorithms," *IEEE International Conference on Evolutionary Computation*, pp. 384-389, 1995.

[10] W. M. Spears, *Evolutionary Programming* IV, The MIT Press, 1995.

Gi Hyun Hwang
He received the M.S. and Ph.D. degrees in electrical engineering from Pusan National University in 1996 and 2000. His research interests include evolutionary computation and fuzzy logic control and their implementation.



Dong-Wang Kim
He received the M.S. and Ph. D. degrees in electrical engineering from Pusan National University in 1987 and 1995. Since 1997, he is an assistant professor of the Dept. of Electricity, Tong-Myong College, Korea. His research interest is robust control.

Hyun-Joon Hwang
He received the M.S. and Ph. D. degrees in electrical engineering from Pusan National University in 1995 and 1999. Since 1998, he is a full-time instructor of the Dept. of Electricity, Pohang College, Korea. His research interest is robust control.

June Ho Park
He received the M.S. and Ph. D. degrees in electrical engineering from Seoul National University in 1980 and 1987, respectively. Since 1985, he has been with School of Electric, Electrical, and Communication Eng., Pusan National University, Korea, where he is currently a professor. His research interests are power system and intelligent control.